Cont A1 and/or C++. However, rule-set development is labor and time intensive, as well as subject to human error and expense.

Please replace the paragraph beginning at page 17, line 18, with the following rewritten paragraph:

A2 A set of object class definitions that can be used by many software applications for the consistent, integrated management of telecommunications and/or data networks and services can be employed. An object class is a definition, or template for a software object that is used to represent physical or logical resources in a software application.

Please replace the paragraph beginning at page 19, line 7, with the following rewritten paragraph:

A3 FIG. 8 depicts a preferred embodiment of the Universal Service Activation Architecture of FIG. 7 that uses artificial intelligence, more specifically, it uses expert system EMS/NMS/OSS 200 for implementing an Universal Service Activation System (USAS) 400 to automatically provision and activate desired/requested service components. Components of FIG. 8 that are similar to those in FIGS. 4, 5 and 7 have been labeled accordingly. The illustrated high-level graphical representation of FIG. 8 is preferably an open, layered operations architecture specifically designed to meet and exceed the architectural needs of large-scale, multi-service networks as they grow in both size and complexity. Universal Service Activation System (USAS) 400 generally includes a Service Provisioning System(s)(SPSs) 402 and an activation system 405 generally comprising, order processing system 406 having a messaging interface(s) 407, order database 356, order processor(s) 375, a peer manager(s) 380, gateway(s) 235, a data archiver 354, Domain Manager(s) (DMs) 410, 415, and 420, and Element Management System(s) (EMSs) 410_1 - 410_n, 415_1 - 415_m, and 420_1 - $420_{k.}$ (The combined Domain Manager(s) and Element Management System(s) 385_1 - 385_N of FIG. 7 are shown separately.) In the depicted FIG. 8, the exemplary Element Management System(s) (EMSs) 410_1 - 410_n, 415_1 - 415_m, and 420_1 - 420_k with corresponding managed network elements $430A_1$ - $430A_n$, $430Z_1$ - $430Z_m$, $432A_1$ - $432A_n$, $432Z_1$ - $432Z_m$, and $434A_1$ - $434A_n$, $434Z_1$ - $434Z_m$, are shown respectively. The order processing system 406 preferably employs a Service Activation Server(s) (SAS) as order processors 375_1 to 375_M substantially similar to the management processor 230 shown in FIG. 5. In other words, the activation system is comprised of service/network management system 200, server/rule

Cont A3

engine 240, and inference base (management information base (MIB)) 250. Although USAS 400 is comprised of software machines stored and executed preferably by a computer, for clarity of the present exposition, those skilled in the art will recognize that activation system components may be stored and executed in different computers/machines.

Please replace the paragraph beginning at page 21, line 28 with the following rewritten paragraph:

A4

With continuing reference to FIG. 8, the SAS 375 generally performs the service management functions, and distribution of service orders to the exemplary Domain Managers 410, 415, and 420 that are managing their respective destination Network Elements (NEs) $430A_1$ - $430A_n$, $430Z_1$ - $430Z_m$, $432A_1$ - $432A_n$, $432Z_1$ - $432Z_m$, and $434A_1$ - $434A_n$, $434Z_1$ - $434Z_m$. SAS 375 makes extensive use of order processing system 406 for the order management process, the palette of objects, intrinsic and rule-sets that support the service provisioning process for access providers and local exchange carriers. The service components flow from order processing system 406 to a collection of DMs 410, 415, 420 through peer manager(s) 380 where such customized interfaces provide the mediation process. Order processing system 406 palettes provide the basis for the following functions. The "External Status Notification" function of order processing system 406 inform external systems of order status, as the order progresses through its life-cycle. With the "Error Propagation function", all errors and reasons are propagated to the external system (in this case the Service Provisioning System(s) 402), if failures occur. The "Persistence" function ensures that the service order supporting data remain in the system as long as required to support rollback and recovery. The "Critical Date Management" function is provided so that the service orders that are in the active state (IN-PROGRESS) are managed within a work queue. A NMS 200 may poll (configureable) to perform periodic evaluation of service orders and their components to ensure that service levels are met. Service Orders that have not yet completed and have exceeded the critical date specified will alarm. These alarms are displayed in the alert display/user interface 245.

Please replace the paragraph beginning at page 24, line 25, with the following rewritten paragraph:

A5

Referring to FIG. 9, the block diagram describes the data flow in one embodiment of the Universal Service Activation System (USAS) 400 which employs the Architecture shown

Cont A5

in FIG. 7. The universal (generic) service component instances 440, 445, and 450 entered in the service provisioning system 402 are grouped together to form a service order 451. This is a sample service order request employing one embodiment of the present invention to illustrate how to construct activation system 405 service order and order administration requests. For example, universal (generic) service component instances 440, 445, and 450 contain component data 452, 454, and 456 respectively. The components can be inter-related and there relationships can be used to describe the order of service activation, and to build more complex services by grouping multiple service components. A specific service component not only describes a specific service but also has a logical ordering with respect to other service components. This ordering applies both to the activation flow and (if required) the backing out (or rollback) of the service component.

Please replace the paragraph beginning at page 27, line 6, with the following rewritten paragraph:

A6

As shown in FIG. 9, a simple service order scenario is included which illustrates the general concept of grouping universal service components into service orders and decomposing service components into specific services or commands supported by the network in one of the embodiments of the present invention. Service components 440, 445, and 450, having component data 452, 454, and 456 respectively of a desired/requested service order 451 may be interpreted, translated, and executed depending upon the nature of the associated data. For example, after decomposition, component data 452, 454, and 456 is sent to appropriate DMs 415 and/or 420. In this particular case, both DMs 415 and 420 receive data. However, after appropriate interpretation, activation system sends component data 454 and 456 to DMs 420 and DMs 415 receives component data 452. In DMs 415 and 420, the received component data is translated in vendor/device specific terminology and sent to targeted network elements $430A_1$ or $430Z_M$ or $432Z_1$ through corresponding EMSs 420_1 or 420_2 or 415_1 respectively.

Please replace the paragraph beginning at page 27, line 25, with the following rewritten paragraph:

A7

The interface A 467 between activation system 405 and service provisioning system(s) 402 can be a generic gateway 235 or via RDBMS table access. It should be apparent to the those having skill in the art that a variety of interfaces can be devised to

Cont A7

communicate between various order creation systems and activation system 405. For example, if the SPSs 402 is based on an underlying database, a DB protocol agent gateway may be employed. To interface to other type of order creation system, a specific type of protocol agent may be needed. In these cases, some custom code may have to be written to forward the order into the activation system gateway and to forward status updates from the activations system gateway back to the order creation system. Further, if the upstream system (customer care or service order entry or service provisioning system) is providing a custom order format to activation system 405 and a NetExpert or NX gateway is used for the interface, then the service order parsing rules must be created. If the upstream system can support the format indicated for activations system 405, then no customization is required.

Please replace the paragraph beginning at page 28, line 14, with the following rewritten paragraph:

A8

The interface B 468 defines the bi-directional communication that occurs between an instance of order processing system and domain managers (DMs). In one embodiment of the present invention, a activation system registry is used to define the service order/request routing and the parameters of the routing. In order for activation system 405 to be aware of the availability of an domain manager (DM) that can serve a specific network elements identified by a network ID, the DM preferably informs activation system 405 the network ID's that it supports. The information is generally kept in activation system registry for later usage in component distributions. Interface C 469 is normally bidirectional and employed to communicate between network elements (NEs) via element management system(s)(EMSs) and domain manager(s)(DMs).

Please replace the paragraph beginning at page 28, line 24, with the following rewritten paragraph:

A9

FIG. 10 indicates a preferred universal service order definition 451 composed of universal (generic) service components 440, 445 , and 450. Service order 451 comprises order- and component-level information. FIG. 10 shows order-level information. The exemplary service order 451 format further comprises of Order begin header 475 to indicate the beginning of a service order, Order end header 476 and Order end statement/command to indicate end of the service order with order number as an attribute. Further the Order header

Cont A9

preferably includes a set of predetermined parameters having a parameter name and a corresponding particular value. For example, in FIG. 10 illustration, following parameters are included: ORDER 478; TYPE 479; TIMESTAMP 481; ACTION 483; RELATED ORDER 485; DATE 487; CRITICAL DATE 489; STATUS 491; OPERATOR 493; ROLLBACK 495 could be assigned manual or automatic; and PRIORITY 497 could be assigned normal, high, expedite or low.

Please replace the paragraph beginning at page 29, line 8, with the following rewritten paragraph:

A10

FIG. 11 shows component-level information for a preferred individual universal (generic) service component 450 definition having component data 456 generally being dependant on service. The exemplary universal (generic) service component 450 format further comprises of Component begin header 510 to indicate the beginning of a component and Order end statement/command to indicate end of the component with order number as an attribute. Further the Order header preferably includes a set of predetermined parameters having a parameter name and a corresponding particular value. Similar to FIG. 10 format, in FIG. 10 illustration, following component related parameters are included: ID 512;SERVICE 514; ACTION 516, NETWORKID 518; CRITICAL DATE 520; PREDECESSOR 522; PREDECESSOR 524; ROLLBACK 526; ROLLBACK 528; PRIORITY 530 could be assigned normal, high or low; and PARTIALALLOW 532 could be assigned yes or no.

In the Drawings

Please replace the drawings of the present application comprising FIGURES 1-15 with the formal drawings submitted herewith.